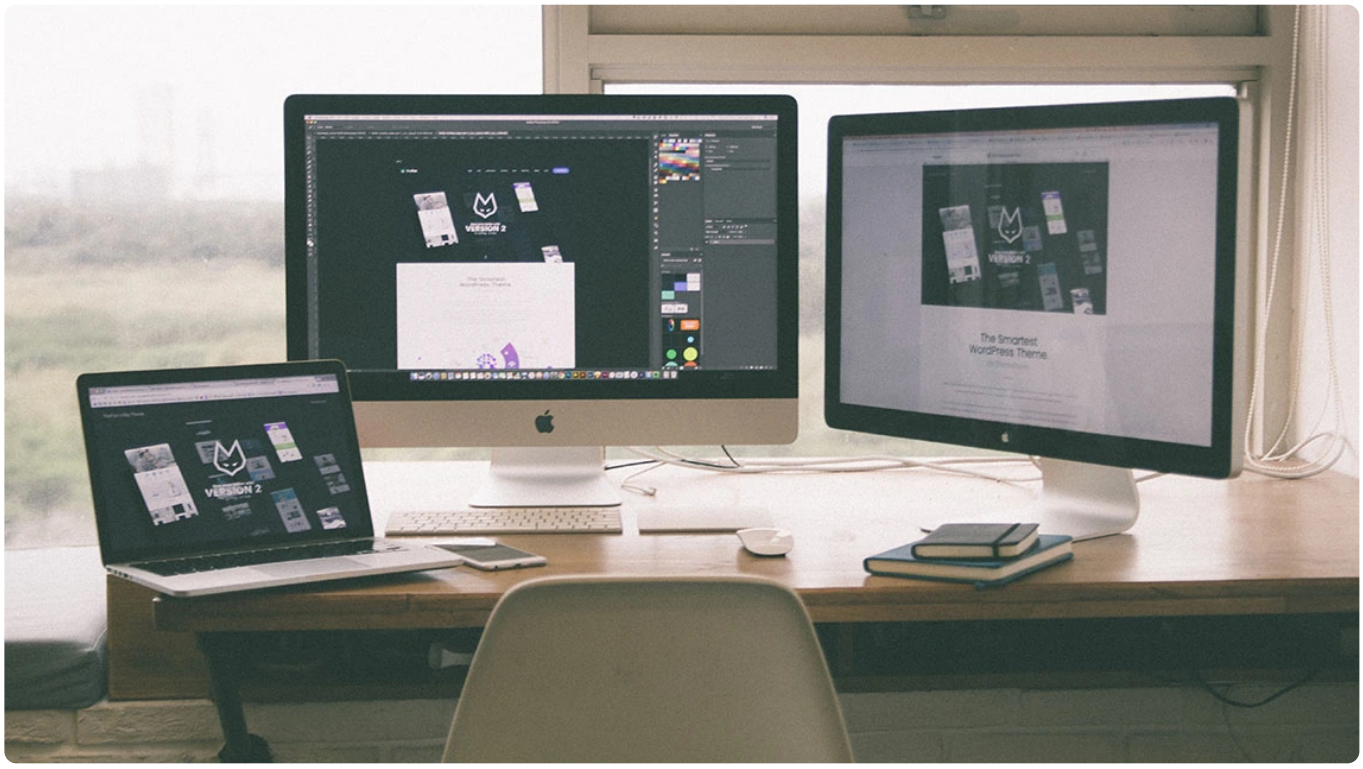
# 11｜网络差怎么办？音频网络传输与抗弱网策略

2021-12-15 冯建元

《搞定音频技术》 课程介绍 >



**讲述：冯建元**

时长 16:26 大小 15.06M

你好，我是建元。

上节课我们讲了如何挑选一个编解码器。其实编解码器就是把音频信息拆解、包装成一个个的数据包，然后通过网络传输到远端。在远端打开数据包，再组装成音频播放出去。

如果把一个个数据包比喻成一辆辆运送音频货物的小车，而网络则是运输的道路。那么我们会发现：有的时候我们经过的是高速公路，物流十分流畅；但有的时候我们经过的却是崎岖蜿蜒的山路，甚至有的小车在山路上，摔下了悬崖，丢掉了包裹，或者赶上交通高峰期道路十分拥堵，从而小车超过了规定的物流时间，这些情况就是我们说的弱网。

一般在弱网情况下，音频的体验可能表现为卡顿、杂音。如果情况严重可能会直接导致无法正常通话。这节课我们就来看看音频链路中弱网是怎么形成的，以及我们是如何通过抗

弱网策略来解决弱网问题的。

## 实时音频传输

在讲弱网之前我们先来看看音频是怎么传输的。在实时音频交互的场景中，为了保证传输的实时性，一般使用基于 UDP 协议的 RTP 协议来传输音频数据。**相较于 TCP 协议，UDP 提供了一种无需建立连接，就可以发送封装的 IP 数据包的方法。**所以它的优点是延迟低、速度快，但丢包了、包损坏了的时候也没有重传机制等做保护，可以说它是一种“尽力而为”的协议机制。

而 RTP 定义了我们音视频的数据包格式，其中包含了 RTP 版本号、包顺序编号等信息。而音频编码得到的压缩后的音频信息，就对应了数据包最后的 Audio Payload，也就是音频负载部分。我们可以通过图 1 来看看一个完整的音频数据包的组成形式。

图1 音频数据包的组成

## 弱网是如何形成的

现在你了解了音频是怎么传输的，接下来我们来看看弱网是如何形成的。其实弱网状态中有三个常见的问题：丢包（Packet Loss）、延迟（Latency）和抖动（Jitter）。我们挨个来看看它们分别是怎么产生的。

## 丢包

我们还是以物流小车为例。“丢包”指的是有的车无法在有效时间内到达终点，甚至可能永远也到不了终点。比如有的小车发生了车祸，或者小车司机罢工了。如果 100 辆车里有 10 辆无法到达终点，那么我们就把它叫做丢包率为 10%。

是的，在**网络传输中，数据包会经过很多复杂的路径，有的是在物理传输中发生了丢失，有的是在服务器、路由转发时由于拥堵或等待时间过长被抛弃。**可以说，互联网传输并不是百分百可靠的，总有数据无法按时传输到目的地。

## 延迟和抖动

在网络这条公路上，从起点到终点我们有很多不同的路径可以选择。你可以选择走高速但也可能走了乡村小道，这样就会导致包裹到达终点所经历的时间发生变化。而这个从发送到接收经过的时间我们把它叫做延迟。

那么很显然，音频在发送的时候是按照时间顺序等间隔发送的，但是由于每个数据包经过的路径不同，从而到达目的地的延迟也不一样。这就导致有的时候很长时间都没有一个数据包到达，而有的时候几乎是同时来了好几个数据包。这就是我们常说的抖动。如果我们按照数据包到达的顺序去播放音频，那么音频播放可能是乱序的而发生杂音，也可能是没有数据可以播放，导致卡顿。

实际上，全球的网络传输环境随区域、时段的不同而不断变化，并且时好时坏。所以丢包、延时和抖动，是基于互联网进行实时传输不可避免的三个问题，不论是在局域网、单一国家地区内传输，还是跨国、跨地区传输，我们都可能会遇到这些问题。

从声网 Agora 监控的网络实况来看，以网络相对较好的中国为例，99% 的音频互动需要处理丢包、抖动和网络延时等。在这些音频会话中，20% 由于网络问题会有超过 3% 的丢包，10% 的会话有超过 8% 的丢包。

## 抗弱网策略

既然弱网状态是互联网传输中不可避免的，那么我们有什么办法来解决或者说对抗弱网呢？主要有网络丢包控制这一网络传输条件下的通用解决方法，和 NetEQ 这种音频独有的抗弱网策略这两块来解决弱网问题。接下来我们逐一看一下。

## 网络丢包控制

我们先来看看我们是怎么对抗丢包的。其实抗丢包背后的逻辑比较简单。简单地说，就是同一个包一次多发几个，只要不是都丢了就能至少收到一个包；又或者丢了包就再重传一个，只要速度够快还能赶上正常的播放时间就可以。这两种思想对应我们通常使用的前向纠错 FEC（Forward Error Correction）和自动重传请求 ARQ（Automatic Repeat-reQuest）这两个纠错算法。

FEC 是发送端通过信道编码和发送冗余信息，而接收端检测丢包，并且在不需要重传的前提下根据冗余信息恢复丢失的大部分数据包。即**以更高的信道带宽作为恢复丢包的开销**。

这里你需要注意的是：音频前向纠错遵循 RFC-2198 标准；而视频前向纠错遵循 RFC-5109 标准。音频由于数据包相比视频要小的多，可以直接用完整的音频包做冗余，而不是像视频用一个分辨率比较差的小数据包做冗余。如图 2 所示这就是 Simple-FEC 的原理。

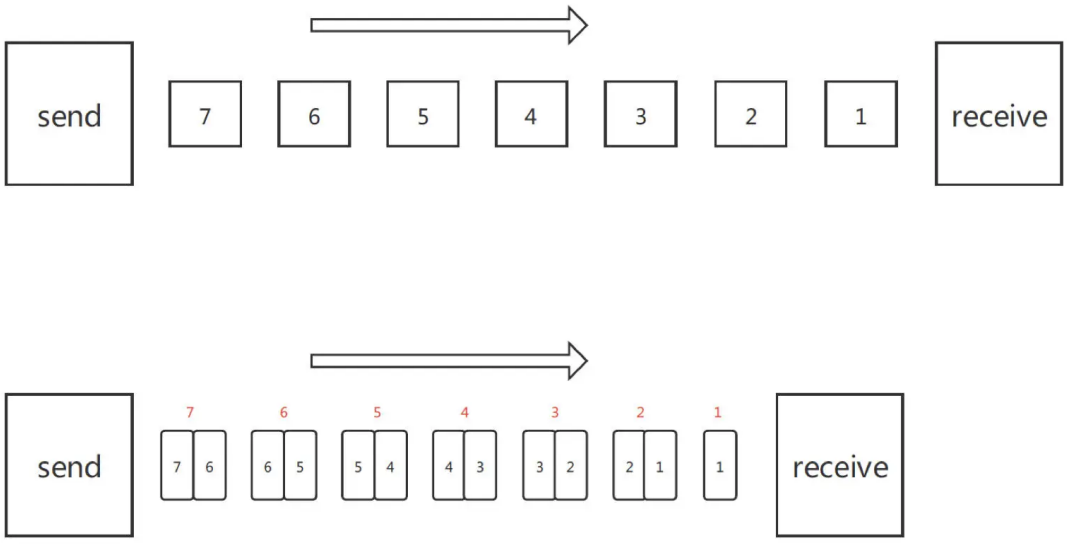



图2 正常发包(上)Simple-FEC(下)

我们看图 2 中的 FEC 就是每次发一个当前时间的数据包和一个上一时刻的冗余包，当其中一个数据包丢失时，我们可以用下一时刻的冗余包把数据恢复起来。显而易见的是，如果同时丢两个包，那么就无法恢复数据了。我们可以看到，在这种多发一倍流量的情况下，连续丢两个包就无法还原了。那么有没有什么办法可以改进这种 Simple-FEC 呢？

让我们再看看另一种 FEC 的方法：RS-FEC，RS 码即里德 - 所罗门码（Reed-solomon Code）。这里我们结合图 3 来看一下。

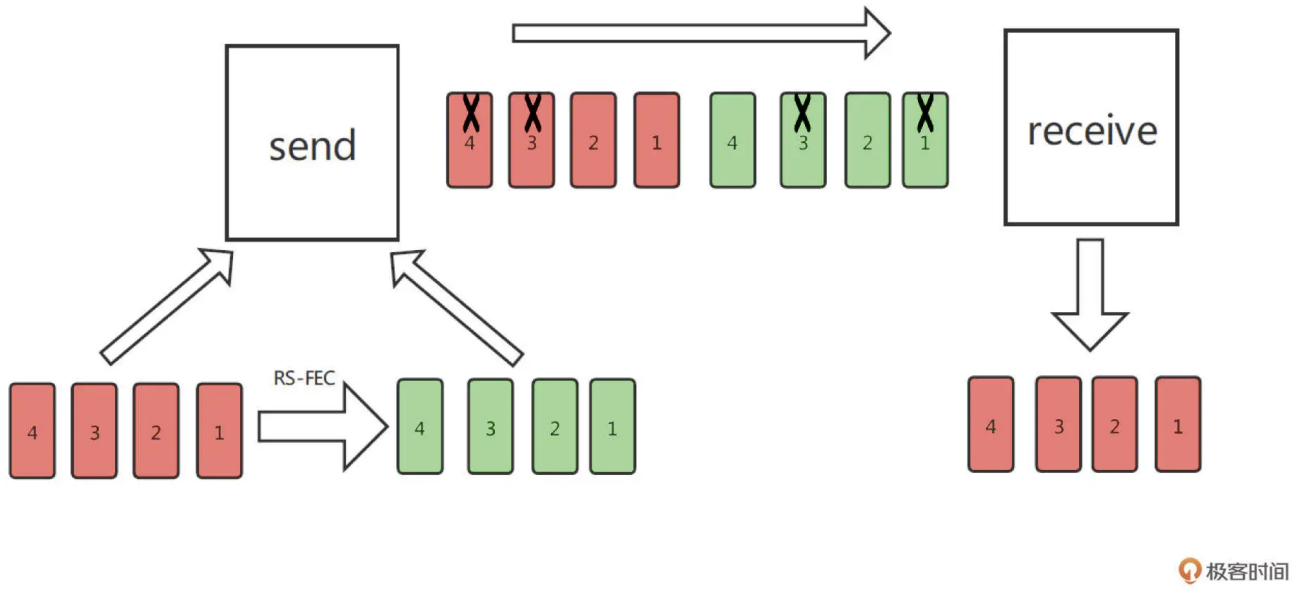



图3 RS-FEC抗丢包原理

我们假设每 m 个包（红色方块）进行一次 RS-FEC 编码得到 n 个冗余包（绿色方块）。冗余包加上原来的包，也就是我们在 m 个包的间隔时间里要发送 m+n 个包。RS-FEC 的特点是，我们只需要得到 m+n 个包中的任意 m 个包就可以把音频还原出来。在图 3 中，m=4，n=4，这样即使这 8 个包里连续丢了 4 个，也就是丢包率是 50%，都可以保证音频的流畅播放。

好的，介绍了 FEC，我们再来看看另一个常用的防丢包策略：ARQ 。其实 ARQ 的原理非常简单。它就是采用使用确认信息（Acknowledgements Signal，ack），也就是接收端发回的确认信息，表征已正确接收数据包和超时时间。如果发送方在超时前没有收到确认信息 ack，那么发送端就会重传数据包，直到发送方收到确认信息 ack 或直到超过预先定义的重传次数。

可以看到相比 ARQ 的丢包恢复，由于 FEC 是连续发送的，且无需等待接受端回应，所以 FEC 在体验上的延时更小。但**由于不管有没有丢包 FEC 都发送了冗余的数据包，所以它对信道带宽消耗较多**。而相比 FEC 的丢包恢复，ARQ 因为要等待 ack 或者需要多次重传。因此，ARQ 延时较大，带宽利用率不高。

我们可以看到，这两种算法都增加了网络所需要传输的带宽。那么如果网络带宽本就不够而导致丢包，这时 FEC 和 ARQ 不但没能起到抗丢包的作用，而且还可能导致网络堵塞导

致丢包更严重。那么除了 FEC 和 ARQ 这两种抗丢包手段外，还有没有什么其它的方法可以解决弱网问题呢？

## NetEQ

其实为了解决弱网问题，在接收端音频解码时通常都有一套比较完整的抗丢包策略。实际上，很多音频编解码器或者开源实时音频框架中都自带了抗丢包策略，其中比较典型的是在 WebRTC 框架中的 NetEQ 模块。我们可以通过图 4 来了解一下。

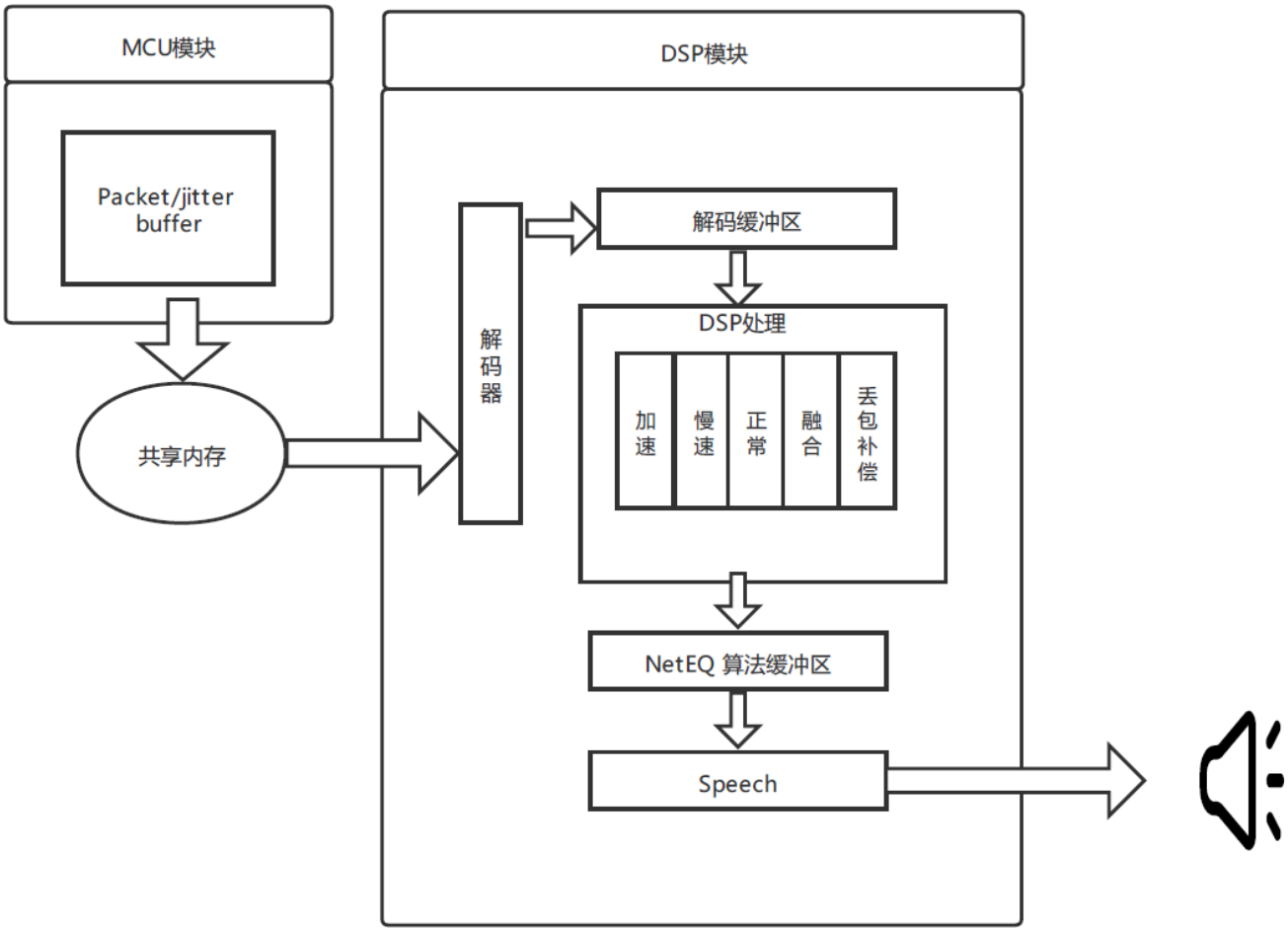



图4 NetEQ处理流程
《实时语音处理实践指南》中国工信出版社

由图 4 我们可以看到，NetEQ 主要包括两个模块：MCU（Micro Control Unit，微控制单元）和 DSP（Digital Signal Prcessing，信号处理单元）。我们知道由于网络传输的不稳定性，虽然我们有 FEC 和 ARQ，但由于延迟或者严重丢包导致的数据包乱序，或者数据包丢失，还是会经常发生的。

在 MCU 里的 Jitter Buffer（抖动缓存区）或者说 Pactet Buffer（数据包缓存区）就是通过开辟一个比较大的缓冲区域，让一段时间内到来的数据包在 Jitter Buffer 里存储、排

序。然后按照播放顺序把数据包交给 DSP 中的解码器进行解码。

在 DSP 模块中，由解码缓冲区得到的音频信号并不是直接交给播放设备播放的。而是需要根据网络状态、缓冲区未处理的数据包长度，以及等待播放的音频长度等参数，来决定使用 DSP 处理中的五种决策方法中的哪一种来处理音频数据。接下来我们就来看看这五种策略：加速、慢速、正常、融合和丢包补偿背后决策的原理、实现方法和实际听感的效果是什么样的。

其实 NetEQ 中主要定义了四种收包的情况：

**1. 过去帧和当前帧都正确收到**

如果过去帧和当前帧都正确接收到了，那么这种情况下只需要考虑网路抖动带来的数据包堆积和数据包接收不足的问题。

所谓数据包堆积，就是同一时间到达了多个数据包都在等待播放，而这个时候需要使用加速策略（accelerate），即对音频信号采用变速不变调的算法来缩短解码后音频的长度，从而实现快速播放。

相反的，如果在缓存中的数据就快播放完了但新包还未送达，那么这时候就需要慢速的方法来把音频时长拉长。这里用到的同样是变速不变调的算法，即只改变音频的播放速度而不改变音频的音调。

WebRTC 中使用的是一种叫 WSOLA 的算法来实现的，这其实是音效算法中变调不变速算法的一种反向应用，更具体地我会在音效算法的一讲中为你详细解读，这里就不再赘述了。

那快慢放的听感是什么样的呢？在网络有抖动的时候，你可能会感觉对面说话，有的时候会快一点，有的时候会慢一点。这种快慢感在语音的时候可能不是那么容易察觉，这是因为人说话本来就有快有慢。但是在音乐的场景下，因为你对一首歌比较熟悉，所以快慢放就会更容易被察觉。

**2. 当前帧发生丢包或者延迟**

如果当前帧发生了丢包或者延迟导致当前没有音频数据可以播放，这个时候就需要额外的 PLC（Package Loss Compensation，丢包补偿）模块来重建音频。你还记得我们在编解码器中讲的 LPC 算法吗？其实常见的 PLC 算法就是通过重建或者复用上一帧的 LPC 系数和残差来还原这一帧的音频数据，从而实现丢包隐藏的。

这里你需要注意和前面的慢放相区分。慢放虽然也可以增加音频的长度但一个慢放系数比例确定后，慢放所能增加的音频长度也就固定了，所以一般慢放用于解决需预测时间比较短的音频的拉长。而 PLC 具有可扩展性，所以一般负责整个一帧或者多帧的，长时间的丢包补偿。

### 3. 连续多帧丢包

那么在连续多帧丢包的情况下依靠 PLC 是不是就可以了呢？答案是否定的，因为 PLC 补出来的音频很大程度上是上一帧音频的延长。如果长时间使用 PLC，声音就会变得失真，从而影响听感。所以如果出现连续多帧丢包，我们就会逐帧递减 PLC 补出音频的能量增益。这也就是为什么，长时间的丢包后的听感是声音逐渐变小直至没有声音，而不是有一个奇怪的声音一直在延续。

### 4. 前一帧丢失，当前帧正常

最后这种情况前一帧可能存在 PLC 的补帧操作，那么新来的音频数据和上一帧就会出现不连续的情况，这里我们就会用到融合的操作。操作也比较简单，就是把当前帧的新数据和之前帧的音频做交叉淡化，让它们的连接处能平稳过度。

交叉淡化的步骤如图 5 所示，其实就是前一帧信号的末尾取一段逐步衰减至 0，然后让后一帧的前端数据从 0 开始逐步提升。然后把这两帧重叠部分相加就可以实现比较平滑的拼接了。

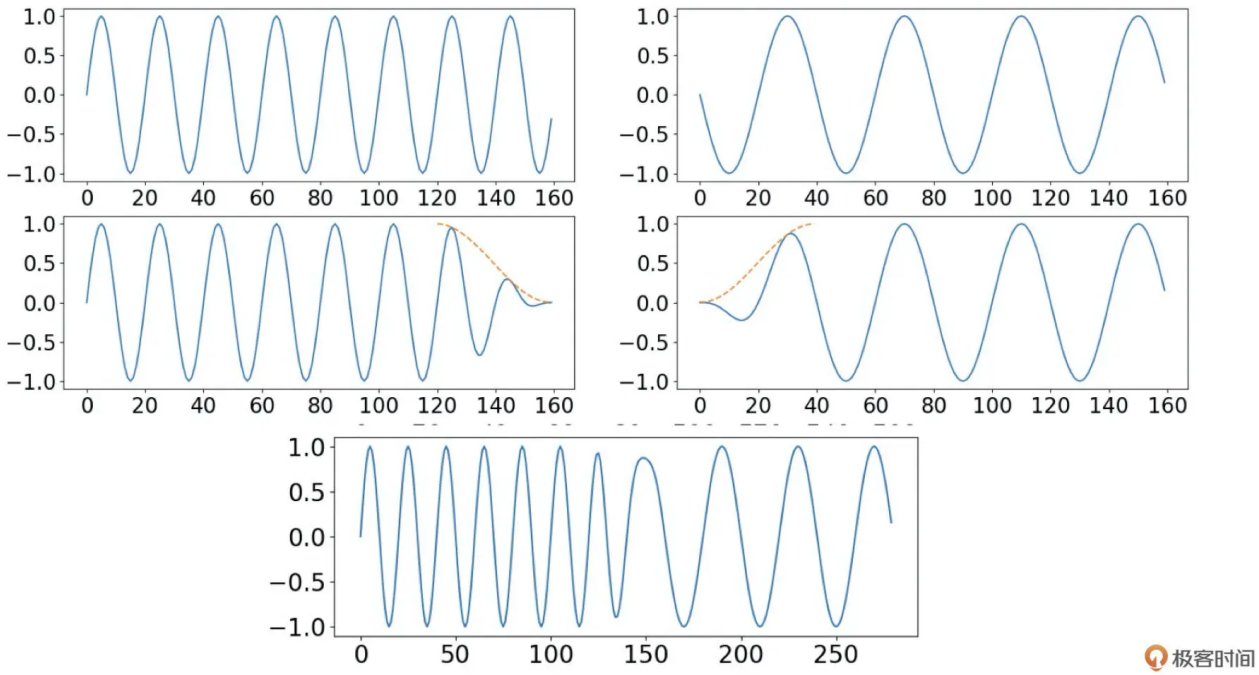



图5 交叉淡化的过程

图 5 中橙色虚线表示交叉淡化用的淡化增益，第一第二行分别表示原始数据和交差淡化衰减后的曲线，最后一行是两帧重叠部分相加、拼接后得到的数据。

## 总结

好的，这里让我们来总结一下这节课的内容。

弱网的情况在实时音频领域中是不可避免的一类问题。它主要表现为丢包、延迟和抖动这三个常见现象。为了能在弱网情况下继续保持音频的流畅播放，我们分别从网络丢包控制这一网络传输条件下的通用解决方法，和 NetEQ 这种音频独有的抗弱网策略这两块来解决弱网问题。

其中，网络丢包控制比较常见的方法有前向纠错 FEC 和自动重传请求 ARQ 这两种。需要注意的是，这两种方法都是以增加网络带宽消耗的方式来提升弱网的能力的。因此，如果在网络带宽本身比较差的情况下，就可能导致适得其反。而 NetEQ 则是通过一些音频处理方法，比如快慢放、PLC 等方法来解决抖动和丢包的问题。

我们可以看到，NetEQ 中通过多个 Buffer 缓存以及快慢放的形式引入了延迟，从而提升了抗网络抖动的能力。然后通过 PLC 的方式解决丢包带来的音频卡顿。这与 FEC 和 ARQ 相比无需额外的带宽消耗，但是却增加了延迟。

实际上抗弱网我们也叫作音频的“最后一公里”(Last-mile)，并且它是保证音频传递的重要组成部分。其实在实际使用中你可能需要针对自己的场景进行一些调整，比如说对于流畅通话比较重要的会议等场景，可以把 NetEQ 中的缓冲 Buffer 适量增大，这样可以进一步提升抗网络丢包的能力。但是 Buffer 也不能太大，这样会导致过多的延迟，从而影响通话效果。

我们也可以在 NetEQ 中引入网络抖动情况的估计，比如在网络抖动严重的时候，动态增加 NetEQ 的 Jitter Buffer 的大小，而网络情况较好的时候减少一些 Jitter Buffer 的大小，从而降低延迟，这些都是可以改进的策略。

## 思考题

这里，给你留一个思考题。在网络带宽比较差的地方，比如只有 3G 网络覆盖的地方，我们要如何调整实时音频系统才能让实时音频获得比较流畅的体验呢？

欢迎你在留言区和我分享你的思考和疑惑，你也可以把今天所学分享给身边的朋友，邀请他加入探讨，共同进步。我们下节课再见。

分享给需要的人，Ta订阅后你可得 20 元现金奖励

生成海报并分享

赞 0 提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 10 | 如何选择一个适合你的编解码器？

下一篇 12 | 空间音频入门：如何实现“声临其境”？

## 精选留言 (4)

写留言

**Leo-J**

2021-12-15

带宽受限可以通过带宽探测嗅探出，这时候应该降低音频的基础码率，比如取消FEC和ARQ等，只让Neteq来发挥作用。

展开

1

**Geek4238**
2021-12-16

感觉你说的都是udp层面的东西，直接用tcp是不是就不存在丢包，顺序问题了?

**running+**
2021-12-15

1）采用G729这种低码率，音质又不太差的这种
2）可以采用FEC和ARQ的融合策略，根据不同情况进行选择不同的抗丢包策略
例如：
a） 如果延时100ms以内，丢包5%以下的可以用ARQ的策略
b）如果延时大于200ms，丢包超过10%以上，可以采用RS-FEC 3+2这种...

展开

**晓龙**
2021-12-15

只有 3G 网络覆盖的地方，我们要如何调整实时音频系统才能让实时音频获得比较流畅的体验呢？

缩减频域范围，降低采样率，取消FEC，ARQ。